Yoga Pose Detection using MoveNet Architecture

Archana Kedar Chaudhari*
Department of Instrumentation Engineering, Vishwakarma Institute of Technology Pune India
E-mail:archana.chaudhari@vit.edu
ORCID iD: https://orcid.org/0000-0002-3304-1461
*Corresponding Author

Praveen Pol
Department of Instrumentation Engineering, Vishwakarma Institute of Technology Pune India
E-mail:Praveen.pol@vit.edu

Srushti Kudtarka
Department of Instrumentation Engineering, Vishwakarma Institute of Technology Pune India
srushti.kudtarkar20@vit.edu

Anish Kulkarni
Department of Instrumentation Engineering, Vishwakarma Institute of Technology Pune India
anish.kulkarni20@vit.edu

Raman Kulkarni
Department of Instrumentation Engineering, Vishwakarma Institute of Technology Pune India
raman.kulkarni20@vit.edu

Rohan Metekar
Department of Instrumentation Engineering, Vishwakarma Institute of Technology Pune India
rohan.methekar20@vit.edu

Abstract: The accurate balance between body parts is crucial in the variety of asanas that yoga has to offer. With the use of yoga-inspired positions, this initiative aims to strengthen the core muscles. Virtual yoga asana practice is possible thanks to the totally accurate position detection provided by the proposed method. To detect exactly the correct position in the proposed method 17 nodes/points are used to denote body parts. The proposed system detects the correctness of yoga poses. Another useful feature used for evaluation of proper pose is heatmap which provides support to detect accurate pose and remove other nonessential parts captured by camera. The proposed work provides an image using React App to guide the user to perform correct yoga pose in real time. The suggested strategy typically achieves real-time speed while maintaining 90% testing accuracy.

Index Terms: Yoga pose detection, Data Preprocessing, MoveNet Model, Heatmaps, key points mapping, React APP

1. Introduction

Yoga, an age-old Indian practice that unites the body and mind, is a wonderful kind of exercise that promotes peace and mental clarity. To get the most out of a specific yoga regimen, a person must adhere to the correct instructions when practicing yoga. As a result, in addition to the regular training sessions led by an instructor, practitioners must devote time to practicing alone. Throughout training, the yoga instructor must adjust each student's posture. One can only practice yoga alone after receiving the right training. But, in the modern world, individuals prefer the convenience of their homes and nearly everything is done online. A technology-driven yoga practice is necessary in this circumstance.

Yoga pose detection has become a topic of increasing interest in recent years as the popularity of yoga continues to grow. Accurate detection of yoga poses can have a variety of applications, including the monitoring of individual practitioners, providing feedback to yoga teachers, and creating interactive yoga experiences. However, despite the growing interest in yoga pose detection, there is still a lack of research in this area. This study aims to bridge this gap by providing a comprehensive overview of existing literature, as well as a proposed method for yoga pose detection using computer vision and machine learning techniques.

2. Literature Survey

Jiaxin Si, Fei Jiang, Ruimin Shen, proposed Large-pose Face Alignment via Shape-aware Heatmap where focus was on dealing with problems of large-pose face alignment [1]. Jothika Sunney, Musfira Jilani, Pramod Pathak, Paul Stynes, proposed a real-time machine learning framework for teaching accurate yoga poses. It combines a pose estimation model, a pose classification model, and a real-time feedback mechanism [2]. Ajay Chaudhari et al. proposed Yog-Guru: Real-Time Yoga Pose Correction System Using Deep Learning Methods [3]. Vinay Chethan et al. proposed methodology working with TensorFlow lite Pose detection python module for recognizing human action based on Yoga Pose Classification using Image Processing and Deep Learning [4]. Xiangjie Huang et al. proposed a system composed of a posture recognition network, a yoga standard movement posture library, a yoga movement correction algorithm and a system UI interface [5]. Silky Goel, et al. presented an overview of various yoga positions and their correct gestures while practicing yoga which could be easily achieved even when one is away from the instructor or yoga tutor for minimizing or prevention of the injuries caused due to incorrect yoga poses [6]. Rao Yingdong et al explored the use of Openpose algorithm to obtain the joint points of the original dataset image, then it connects the different joints together appropriately, to obtain the skeleton image [7]. P. Paula and S. William, presented programs that span a wide variety of genres, platforms and input technologies, from the touch screen of a smart phone to the full-motion, the natural input of devices like the Kinect Sensor [8]. Jinyoung Jun et al. proposed a novel skeletal attention module to generate keypoint heatmaps, which exploits skeletal, as well as overall body structure, information for human pose estimation [9]. Yimin Zhou et al. paper presented a high-level hand feature extraction method for real-time gesture recognition. Firstly, the fingers are modelled as cylindrical objects due to their parallel edge feature [10]. Rishabh Bajpai and Deepak Joshi, presented a Deep NN for knee joint profile prediction at variable walking speeds and slopes for able-bodied users. This proposed model is trained and verified on a public data set having data of 270 min recorded from ten subjects. First, it can reduce the gait data recording time as data recorded from the flat surface can predict gait data for other AOIs. Second, it can propose continuous reference trajectories to an exoskeleton during variable walking speeds and slopes [11]. Zixin Cai et al. proposed Pose Estimation Workout Mobile Application named as PoseBuddy, which creates an interactive mode that collects users' poses using the front camera of the mobile device and inputs them into a high-accuracy human pose estimation model made available by TensorFlow, MoveNet. After being validated by the accuracy validation method, it then offers real-time auditory feedback while exercising. Users may see in real-time whether they are performing the exercise properly because of the system's ability to record and interpret real-time inputs from the camera stream. PoseBuddy is also a sports site where users may share their experiences after working out and make new acquaintances [12]. The joints positions of the body segments are found using the MoveNet pose estimation algorithm. Then, in order to examine how they vary, the knee and thigh-trunk angles on both the left and right side are calculated in each frame of the video sequence. The analysis will be done at the stride level by using statistical factors because the video sequences can have a variety of lengths and frame rates. A smaller sample of film sequences with both pathological and normal gait were used in the experiments. To boost the amount of data used for gait classification, data augmentation was used [13]. G. Sathyanarayanan et al. presented a study on the possible advantages of yoga and mindfulness in treating serious mental diseases. Along with their underlying mechanisms, it addresses how they affect symptom intensity, cognitive function, and psychosocial functioning. To determine their effectiveness in treatment programmes for serious mental diseases, more research is required [14]. H. T. Chen et al. presented a review of the creation of a system for the automatic recognition of yoga postures using computer vision techniques. The authors provide an explanation of their system's feature extraction, feature selection, and classification techniques. The study analyzes difficulties in identifying yoga poses, offers experimental findings, and emphasizes the potential of their approach for yoga self-practice. It is also recommended to include real-time feedback and customizable tools in future studies [15]. M. U. Islam explored and presented a procedure for locating joint points, including joint point extraction, posture recognition algorithms, and data pre processing. The authors give experimental findings that show how their suggested system recognises various yoga postures with accuracy and performance in real-time. The report also examines the system's shortcomings, including susceptibility to occlusion and differences in body types and attire. It emphasizes the system's potential for use in providing real-time feedback and direction during yoga practice, as well as the use of technology to support good lifestyle choices [16]. S. K. Yadav et al. presented deep learning method for real-time yoga posture detection. The authors' proposed pose recognition using convolutional neural networks (CNNs), with an emphasis on data gathering, pre processing, and CNN training. While noting limitations and difficulties in recognising yoga postures, the study covers experimental findings and highlights the possible uses of their system in real-time feedback, assistance, and monitoring during yoga practice [17]. Y. Chena, Y. Tianb and M. Hea, give an in-depth analysis of deep learning-based approaches for estimating monocular human pose. An overview of current developments in this topic, including different network architectures, data representation methods, and training procedures, is given in the study. The study covers the benefits and drawbacks of current approaches and outlines potential future lines of investigation in the area of monocular deep learning human posture estimation [18]. D. Osokin, a simplified version of the OpenPose system for CPU-based real-time 2D multi-person posture estimation. The strategy, which included model architecture, network pruning, and quantization techniques, is described in detail as it was applied to the original OpenPose system to make it more effective for CPU-based inference. The suggested Lightweight OpenPose system performs well in real-time on a CPU, as shown by experimental results in the paper, making it appropriate for use in practical applications with constrained computational resources [19]. H.-T. Chen et al. described a computer-assisted yoga training system that makes use of computer vision techniques for in-the- moment yoga posture recognition and feedback. The system's architecture, which comprises modules for pose detection, feature extraction, and posture recognition, is described in the paper. The system's capabilities, constraints, and prospective applications for enhancing yoga practise through real-time feedback and direction are also covered by the authors. The study also includes experimental findings and a performance assessment of the system [20]. Human poses and motions are crucial cues for the analysis of human-centered movies, and there is compelling evidence that body pose-based representations are very good at a wide range of tasks, including activity detection, information retrieval, and social signal processing. By creating "PoseTrack," a new large-scale standard for video-based human posture estimation and articulated tracking, and by bringing together the community of researchers working on visual human analysis, we want to enhance the state of the art. The benchmark includes three competition tracks that concentrate on (i) multi-person pose estimation in single frames, (ii) multi-person pose estimation in videos, and (iii) multi-person articulated tracking [21]. W. Yang, W. Ouyang, H. Li and X. Wang, proposed End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. Deep Convolutional Neural Networks (DCNNs) have recently been used for the job of estimating human position, and they have demonstrated their capacity to develop better feature representations and capture contextual linkages. However, it is challenging to incorporate domain-specific previous information into DCNNs, such as geometric correlations between body parts. Additionally, ambiguities are introduced during training of DCNN-based body part detectors when global body joint consistency is not taken into account. Using DCNNs and the expressive deformable mixture of components, we present a unique end- to-end framework in this research for estimating human pose [22]. L. Pishchulin et al. presented Deepcut: Joint subset partition and labeling for multi person pose estimation. In this paper, the challenge of estimating numerous people's articulated poses from photographs of the real world is discussed. They suggest a method that combines the tasks of detection and position estimation. It estimates the population of a scene, recognizes obscured body parts, and distinguishes between body parts of individuals who are physically close to one another. In contrast to earlier approaches, this joint formulation tackles the issue by first recognising persons and then assessing their body posture. We offer a formulation for partitioning and labeling a set of body- part hypotheses produced by CNN-based part detectors. In our formulation, which is an example of an integer linear program, non-maximum suppression is implicitly performed on the set of part candidates before they are grouped to create configurations of body parts [23]. H. E. Downs, et al. proposed, the effectiveness of using video self-evaluation and video feedback techniques to improve yoga position accuracy was assessed. Participants in the self-evaluation intervention rated task analyses for feedback on each position while viewing video of their behaviors. The task analyzes for the behaviors as well as practice during video recordings were rated by the participants and the researcher jointly in the video feedback intervention. A design with numerous baselines across behaviors was used to evaluate the interventions. In accordance with the findings, both video self-evaluation and video feedback improved the accuracy of every posture for one participant [24].

3. Proposed Methodology

The proposed work is divided into two sub-sections. The initial sub-section describes the MoveNet architecture for yoga pose detection. The next subsection elaborates how real time implementation and tracking of Yoga poses can be performed using React Web App.

A. Proposed MoveNet Model for Yoga Pose Detection

MoveNet is a very quick and precise model that can identify 17 important bodily parts. Fig. 2 describes the 17 Key Points from Active keypoint dataset. Two variations of the device, dubbed Lightning and Thunder, are available on TF Hub. Thunder is designed for applications requiring high accuracy, while Lightning is designed for applications where latency is essential. For live fitness, sports, and health applications, both models just operate faster than real time (30+ FPS) on the majority of contemporary desktops, laptops, and smartphones.

Fig. 2 Seventeen Key Points from Active key point dataset

In the last five years, human pose estimation has advanced significantly, but surprisingly, it hasn't been used in many apps just yet. This is due to the fact that bigger and more accurate pose models have received more attention than work on making them quick and deployable everywhere. MoveNet is a bottom-up estimate technique that precisely locates human key points using heat maps. A feature extractor and a group of prediction heads make up the architecture. The prediction technique roughly follows CenterNet, although there are some significant modifications that enhance both speed and accuracy. Using the Tensor Flow Object Detection API, all models are trained. A high resolution (output stride 4) and semantically rich feature map are produced by MoveNet's feature extractor, MobileNetV2, which has an attached feature pyramid network (FPN).

i) Pre-processing steps for use of Movenet Model

Preprocessing is crucial for the MoveNet model's ability to detect numerous postures in yoga asanas since it guarantees the excellent quality of the training data. This increases the model's precision and keeps it from overfitting the training set of data. By removing noise, artifacts, and inconsistencies from the data, preprocessing techniques including picture cleaning, enhancement, normalization, and background removal serve to strengthen and increase the reliability of the model. Following are the pre-processing procedures for utilizing the MoveNet model to detect numerous poses and other asanas:

1. Image Collection: Here we collected a large number of high-quality images of people performing the asanas in various settings, such as indoors, outdoors, with different clothing, and under different lighting conditions. The images should be taken fromdifferent angles and perspectives to capture the variations in the poses.

2. Image Cleaning: This step is important to Clean the images by removing any unwanted noise or artifacts that may be present in the images. This can be done by using techniques such as denoising, removing shadows, and removing background clutter.

3. Image Resizing: Resize the images to a standard size that is compatible with the input size of the MoveNet model. The MoveNet model requires images with a resolution of 256 x 256 pixels, so it's important to resize the images to this size to ensure consistency in the input images. We made use of the popular image resizing library OpenCV to resize the images.

4. **Pose Annotation:** Annotate the images with the corresponding pose labels. This involves marking the key points of the pose, such as the joints, limbs, and body parts. Here, tools like LabelIng or VGG Image Annotator (VIA) are suitable for this task. It's important to make sure the annotations are accurate and consistent to ensure the model learns the correct features.

5. Data Augmentation: Next step is to Augment the dataset by applying transformations such as rotation, scaling, and flipping to the images. This will increase the variety of poses in the dataset and make the model more robust. Using the most popular image augmentation library Keras Image Data Generator to perform data augmentation.

6. Splitting the dataset: Divide the dataset into training, validation, and test sets. The training set is used to train the model, the validation set is used to tune the hyper parameters of the model, and the test set is used to evaluate the performance of the model. The split should be done in a way that ensures that there is no overlap between the images in the different sets.

7. Normalization: Normalize the pixel values of the images so that they have a mean of 0 and a standard deviation of 1. This can help the model to converge faster during training. Here, making use of normalization techniques like Z-score normalization or min-max scaling.

8. Background Removal: The MoveNet model is designed to detect poses on a plain background, so it's important to remove any background clutter from the images. This can be done using background removal techniques such as thresholding, morphological operations, or machine learning-based methods.

9. Frame Extraction: Finally, Extracting individual frames from videos of people performing the asanas. This was done using video processing libraries like OpenCV. By extracting frames, you can create a larger dataset with more variations in the poses.

By following these pre-processing steps clean and varied dataset that can be utilized to train a reliable pose detection model using the MoveNet architecture.

ii) Heatmap Visualization for exact mapping of keypoints for pose detection

According to its definition, heatmap visualization, also known as heatmap data visualization, is a way to graphically depict numerical data in which the value of each data point is denoted by a different color. The warm-to-cool color scheme, which uses warm colors to represent high-value data points and cool colors to indicate low-value data points, is the one that is most frequently used in heatmap visualization. Fig. 3 demonstrates the heatmap and key point relation.

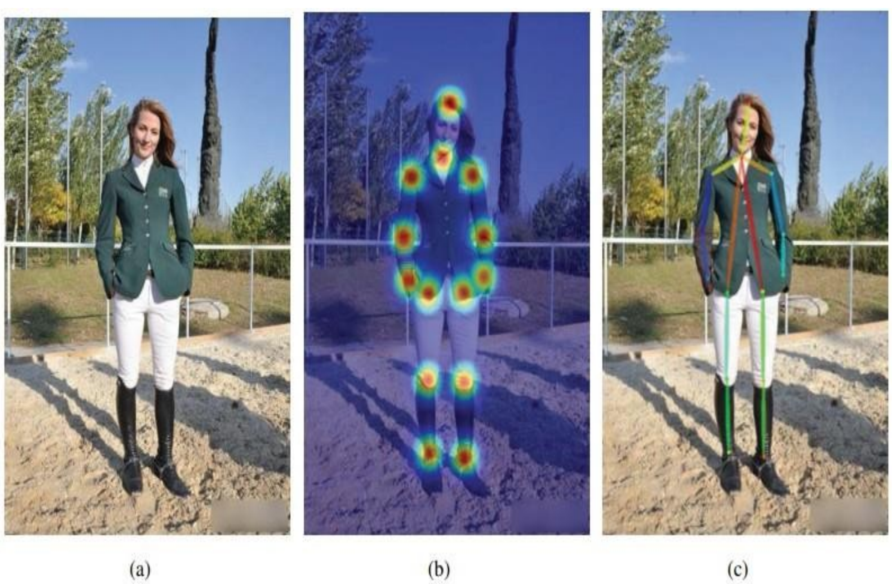


Fig. 3 Heatmap and key point relation: A single person pose estimation SPPE Model using heatmap regression a) original image b) heatmap generated c) detection result (Courtesy: TensorFlow official)

In order to create heatmaps, a color gradient is often applied to an image or a 2D plot, with the color intensity corresponding to the magnitude of a given variable. Heatmaps can be used, for instance, in pose detection to highlight the areas of an image that correspond to various body parts, like the head, shoulders, and knees. The keypoint distribution is represented smoothly and continuously in the heatmaps by the use of a Gaussian filter centered at the keypoint positions. A variety of methods, such as color maps, contour lines, and surface plots, can be used to visualize heatmaps. The most popular visualization technique uses color maps, where the intensity of the color corresponds to the size of the variable the heatmap is intended to depict. The steps for using heatmaps to detect poses are as follows:

1. Image Input: The first step is to enter a picture of someone doing Yoga asana. Then, using the previously covered preprocessing methods, this image is resized, normalized, and the background is removed.

2. **Model Prediction:** The MoveNet model is then applied to the preprocessed image to get the anticipated poses. The model generates a set of keypoint coordinates that reflect the positions of different body parts, including the hips, knees, shoulders, elbows, and ankles.

3. Heatmap Generation: Using the keypoint coordinates, a heatmap is then produced that emphasizes the areas of the image that correspond to each body part. A Gaussian filter centered at each keypoint position is used to create the heatmap. The heatmap that is produced has an intensity that is stronger at the keypoint position and gradually lowers as it moves outward.

4. **Pose Estimation:** After that, the heatmaps are utilized to determine how the person in the picture is posed. Finding the links and calculating the angles between the body sections allows for this. The estimated pose is determined by selecting the pose that matches the angles closest to the expected angles for each stance.

5. **Visualization:** The estimated pose is shown using heatmaps in the last stage. The location and intensity of each keypoint are displayed as heat maps overlaid on the input image. In addition to giving a clear and simple visualization of the position, this also makes it easier to spot any mistakes or flaws in the model's output.

In conclusion, heatmaps are an effective technique for pose detection of numerous poses in Yoga asanas because they offer a means to view and evaluate the model's output. For applications like fitness tracking, sports analysis, and physical rehabilitation, accurateand dependable pose assessment is made possible by the usage of heatmaps.

iii) Keypoints Mapping

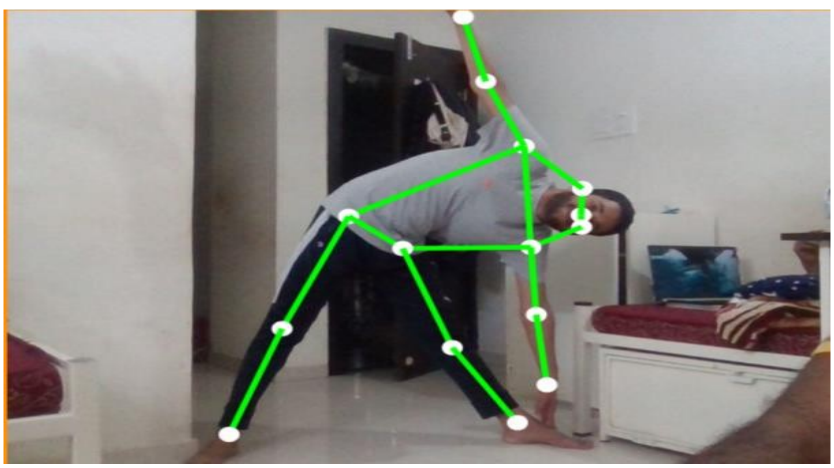


Fig. 4 Key point mapping

Following are the steps for keypoints mapping

Step 1: To find the centers of each person in the frame, the person center heatmap—which is the mathematical mean of all key points related to a person—is used. The region with the greatest number is selected, with ranking based on distance from the frame's center.

Step 2: To generate the very first set of key points for the person, the keypoint regression output is cut into the pixel that matches the object center. This is a middle-out prediction which requires work across multiple scales, in order that the quality of regressed keypoints won't be highly accurate.

Step 3: Each pixel in the keypoint heatmap is amplified by a value of weight which is inversely correlated with its distance from every regressed keypoint. It also makes sure that we refuse to include key points from background subjects because they generally won't be close to the key points getting regressed and as a result will have inadequate outcomes.

Step 4: The last collection of keypoint forecasts is selected by obtaining the locations of the greatest heatmap values for every keypoint channel in step 4. To generate accurate projections, these values are next appended to the area 2D offset projections.

Fig. 4 presents the accurate keypoint mapping. The key points in the Fig. 7 and overall project are - Nose,Right Ear,Right Shoulder,Right Elbow,Right Wrist,Left Shoulder,Left Elbow,Left Wrist,Right Hip,Right Knee,Right Ankle,Left Hip,Left Knee,Left Ankle,Right Eye,Left Eye, Right Ear.

iv) Movenet Architecture using TensorFlow

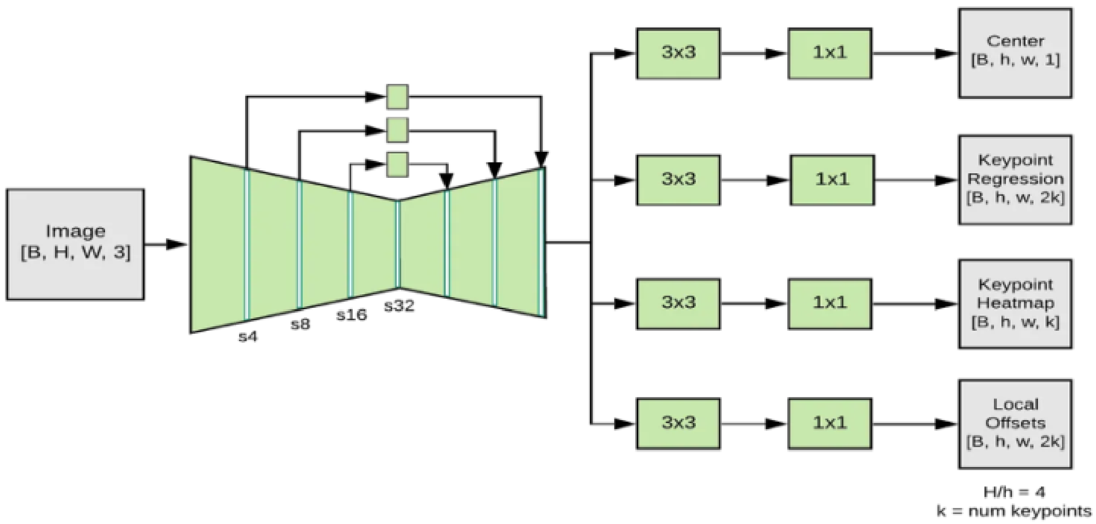


Fig. 5 Movenet Architecture (courtesy: TensorFlow official)

In order to obtain an increased accuracy of the Movenet model, the proposed work uses CNN as the initial layer for training. We build a CNN model that consists of two convolutional layers, a max-pooling layer, and a fully connected layer as the output layer's final layer. The specification of the model is that implements for image classification for first five poses initially we intended to implement without early stopping or dropout layers.

It utilizes convolutional layers with 3x3 kernels and ReLU activation to extract features, followed by max pooling layers for down sampling. The output is flattened and passed through a fully connected layer with softmax activation for class probability prediction. The model is compiled with the Adam optimizer, using a learning rate of 0.0001, and categorical cross-entropy loss.

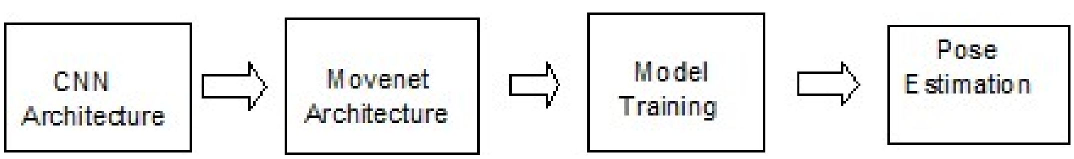


Fig. 6 Block Diagram of the proposed methodology for pose detection

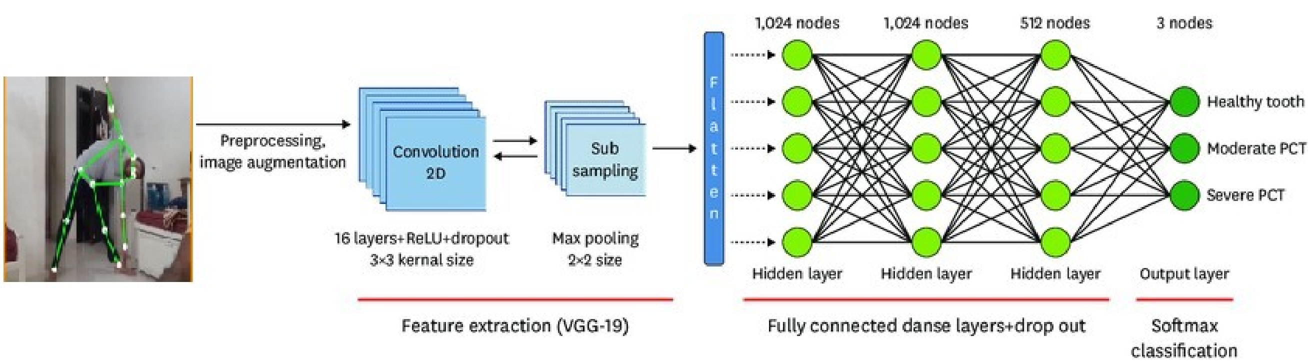


Fig. 7 CNN Architecture [25]

Accuracy is used as the evaluation metric during training, which runs for 10 epochs. The model's predictions on the validation dataset are compared to true labels, generating a confusion matrix to assess classification performance. Fig. 5 presents the Movenet architecture. Fig. 6 represents the block diagram of the proposed method for pose detection. Fig. 7 demonstrates the CNN architecture used in the proposed model.

B. Real time pose detection and tracking using React Application

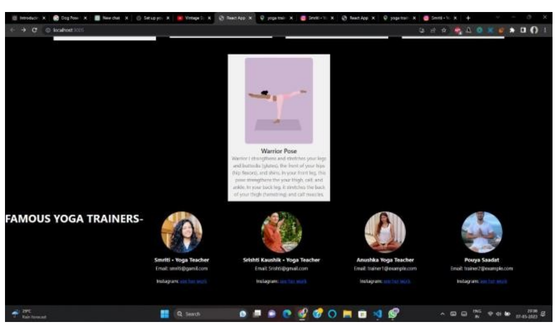
For the given proposed classification model, a React application for the real time use of the proposed model is created. The app module are About, Home, Tutorials and Yoga as shown in Fig. 8.

Fig. 8 Pages created for the app

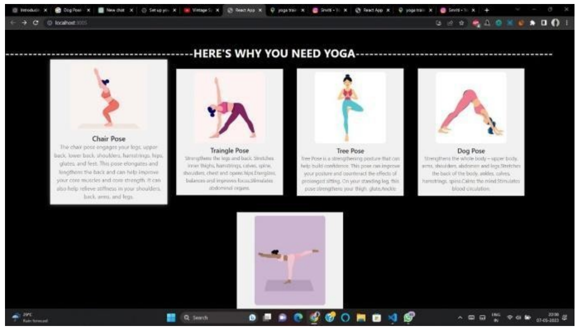
For the scripting purpose, JavaScript was used and CSS was used for editing and styling purposes. Following are the features of different pages created in the App. Currently the App is hosted on local Host.

Fig. 9 represents the various pages of the Web App along with the mentioned feature. Fig. 10 presents the real time interface for yoga pose detection.

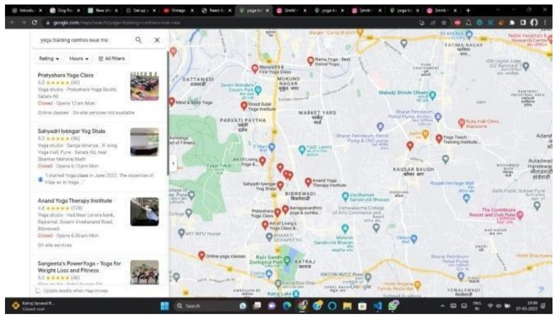
(a) Home Page



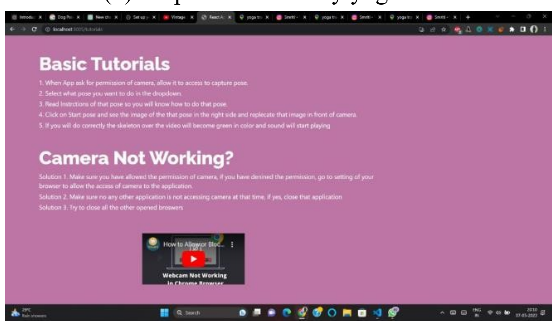
(b) Information about famous yoga trainers



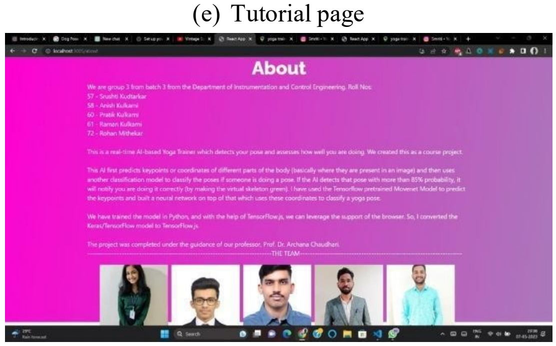
(c) Different yoga asanas and their benefits



(d) Maps to find nearby yoga tutor

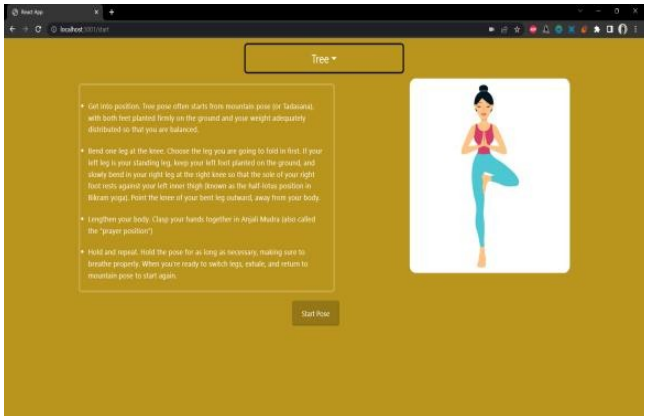


(e) Tutorial page

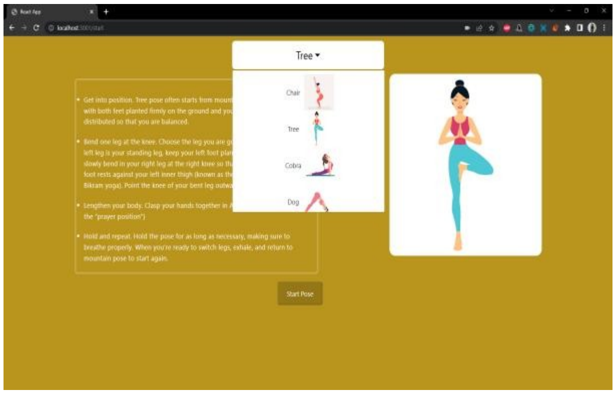


(f) About page of the Web app

Fig. 9 Web Applications for real time yoga pose detection and various pages



(a)



(b)

Fig. 10 Real time interface for yoga pose detection using React App

4. Results and Discussion

A. Dataset Description

In the suggested project, we created a dataset of seven yoga poses, each with 80 to 100 images obtained from YouTube, Google, and other sources, and divided it into training and testing sets in a ratio of 75:15. Fig. 11 demonstrates the snapshot of stored database.

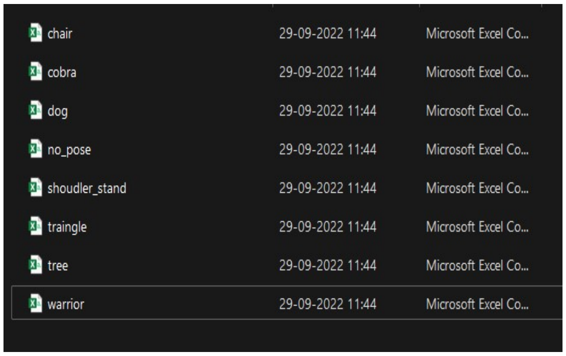


Fig. 11 Snapshot of the stored Dataset

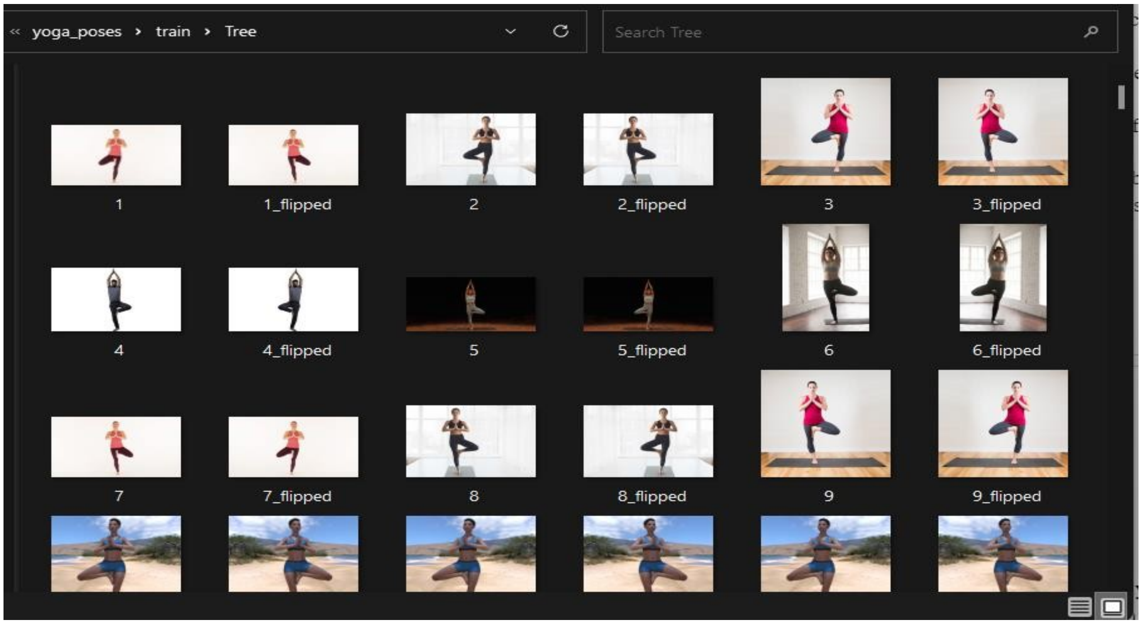


Fig 12. Train Set for **Tree Pose**

Standing posture known as the **Chair Pose e**ntails lowering the hips and bending the knees to simulate sitting in a chair, while raising the arms aloft. **Cobra Pose a** prone position in which the hands are put on the floor beneath the shoulders and the upper body is lifted off the floor while the lower body remains on the floor. Standing pose known as the **Downward Dog** entails putting the hands and feet on the floor while lifting the hips up and back to form an inverted V-shape with the body. Savasana or seated meditation is examples of poses that could be referred to as **No pose**.

In the supported inversion position known as the **Shoulder stand**, the legs are raised above the head while the body is supported by the hands on the back. **Triangle Pose a** standing position in which the front leg is crossed over the rear leg, the arms are extended to the sides, and the torso is positioned over the front leg. Standing in **Tree Pose** one balance on one leg, place the foot of the other leg on your inner thigh or calf, and raise your arms aloft. In the standing position known as the **Warrior Pose** the arms and legs are extended to the sides or aloft while one leg is lunging forward and bent at the knee. Fig. 12 elaborates the Train set of **Tree Pose**.

B. Performance Evaluation and Metrics for Comparison with other methods

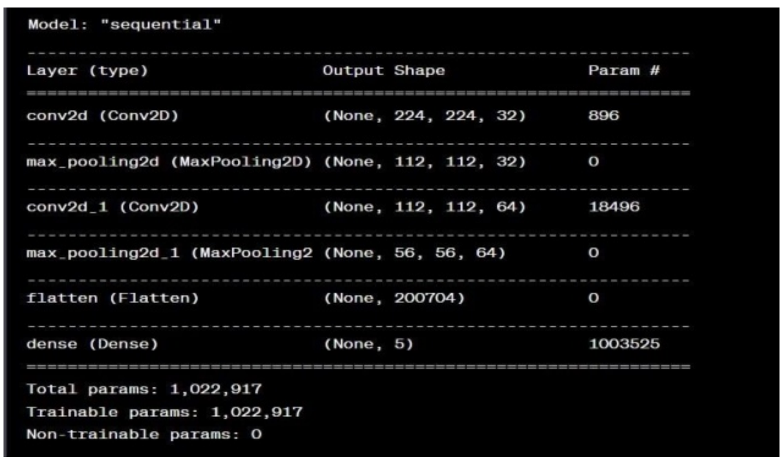
Metrics like accuracy, precision, recall, true positive (TP), and false positive (FP) are frequently used in the fields of machine learning and data science to assess the effectiveness of classification models. How frequently the model predicts an observation's class accurately is known as **accuracy.** The amount is determined by dividing the total number of forecasts by the number of accurate ones. Among all positive predictions, precision counts the percentage of true positives. By dividing the total number of anticipated positives by the number of actual positives, it is computed. Measured by recall, the true positive rate is the percentage of all real positives. By dividing the total number of real positives by the number of true positives, it is computed. Table 1 describes the results of comparison of proposed method for yoga poses detection using Movenet and other methods.

Table 1: Performance comparison of different posed detection algorithms

Sr.no	**Name of the algorithm**	**Dataset used (Yoga poseused for training)**	**Performance metrics (Accuracy, Precision,Recall, TP,FP)**
1	Proposed method usingCNN+ Movenet	Own dataset using yoga poses available on Google	Accuracy: 97% forpose identification
2	CNN - Convolutional Neural Networks [3]	Data-set of Five Yogapose	Accuracy: 95% forpose identification
3	TensorflowLite, DeepLearning [4]	Five different people did Surya Namaskar poses	Accuracy:98%
4	Star skeleton Computation [15]	Yoga video collected	Accuracy: 99%

C. Results and Discussions

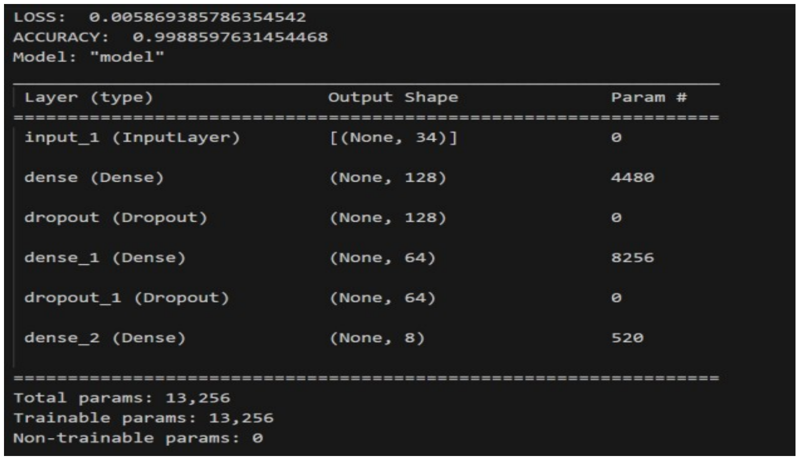
Fig. 13 presents summary of the CNN model.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 224, 224, 32)      896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 112, 112, 32)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 112, 112, 64)      18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 56, 56, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 200704)            0
_________________________________________________________________
dense (Dense)                (None, 5)                 1003525
=================================================================
Total params: 1,022,917
Trainable params: 1,022,917
Non-trainable params: 0
```

Fig. 13 Summary of the CNN model

Fig. 14 presents the implementation of the Movenet model. Fig. 15 demonstrates the accuracy plots for the detection of Yoga pose using Movenet architecture. Fig. 16 presents the confusion matrix and Fig. 17 represents the Accuracy and loss plots while model training. The model demonstrates the following results, Training Accuracy = 0.92, Validation Accuracy = 0.94 and Testing Accuracy = 0.97.



```
LOSS:  0.005869385786354542
ACCURACY:  0.9988597631454468
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 34)]              0

 dense (Dense)               (None, 128)               4480

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 64)                8256

 dropout_1 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 8)                 520

=================================================================
Total params: 13,256
Trainable params: 13,256
Non-trainable params: 0
```

Fig. 14 Snapshot of the Movenet model

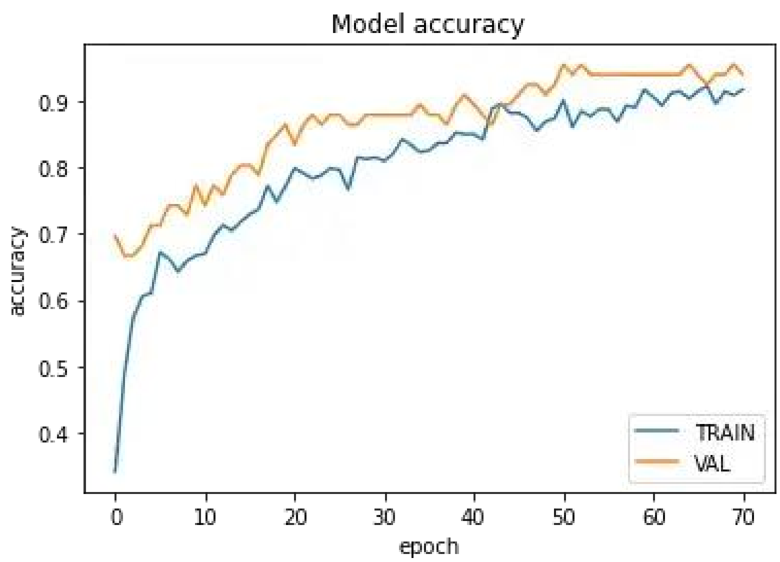


Fig. 15 Accuracy plots for the detection of Yoga pose using Movenet architecture

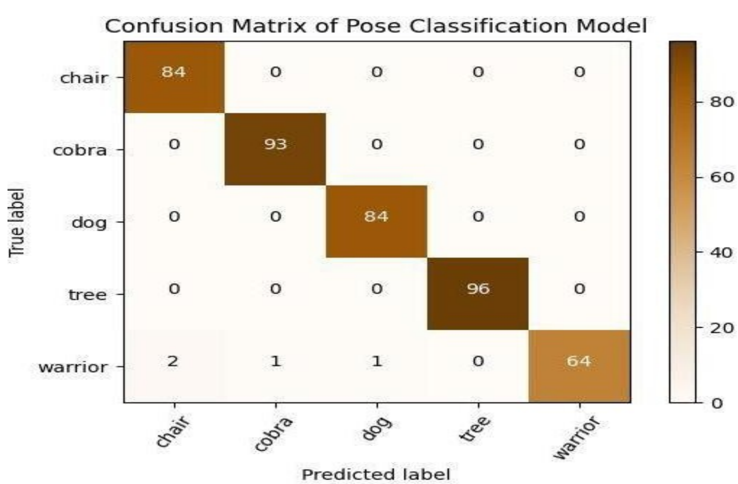


Fig. 16 Confusion matrix results for First 5 poses

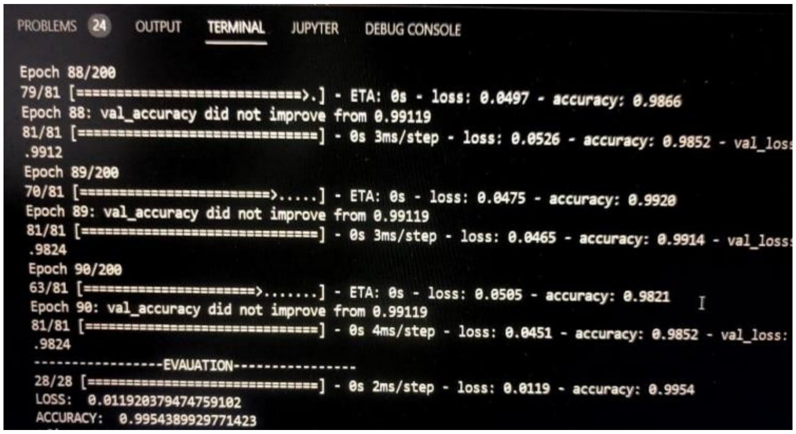


Fig. 17 Snapshot of Accuracy and Loss

Fig. 18 demonstrates the real time implementation of yoga pose detection and tracking using the React App for Triangle pose. From Fig. 13, if pose is correct then our 17 Key Points will be green color then pose will be correct and pose time will start , we can see our best pose time. If pose is incorrect the lines between the keypoints will be in white color and pose timer in the Appinterface will not start.

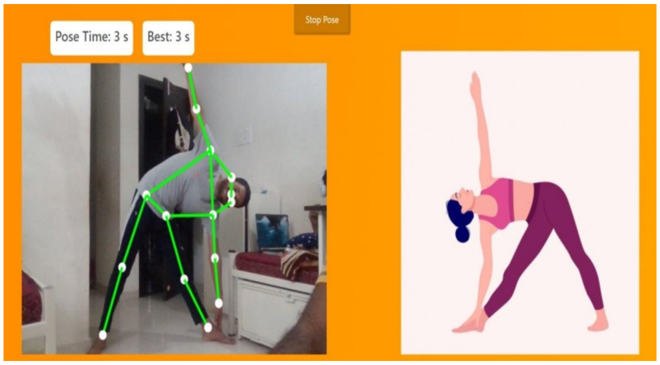


Fig.18 Real time detection and tracking of **Triangle Pose** (Left) real time yoga pose (Right) AccurateTriangle pose used for mapping the keypoint for tracking corrections

Fig. 19, 20 and 21 exhibits the real time implementation of yoga pose detection and tracking using the React App for Chair pose, Warrior pose and Dog pose respectively.

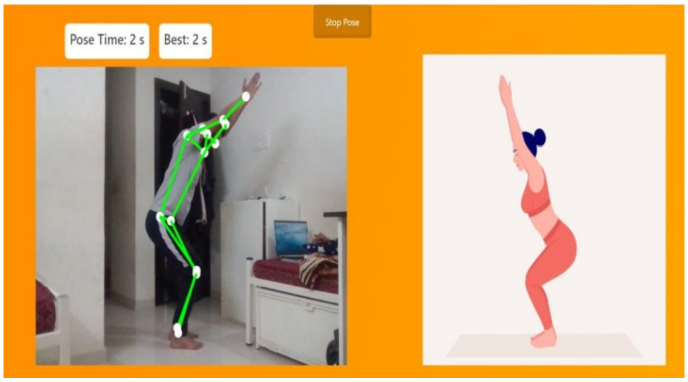


Fig.19 Real time detection and tracking of **Chair Pose** (Left) real time yoga pose (Right) image ofcorrect Chair pose used for mapping the keypoint for tracking corrections

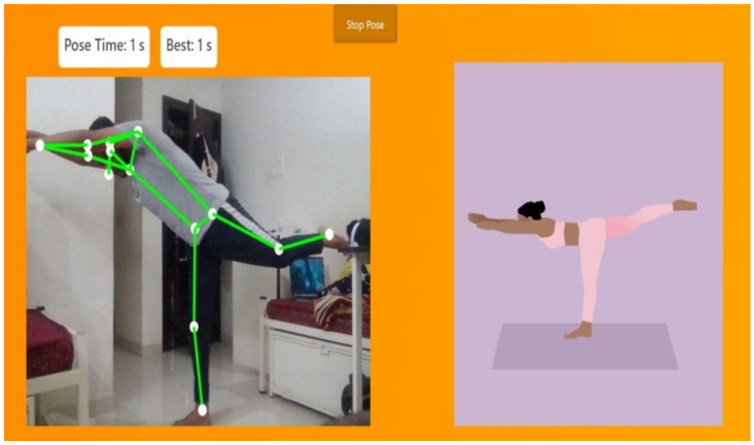


Fig. 20 Real time detection and tracking of **Warrior Pose** (Left) real time yoga pose (Right) image ofcorrect Warrior pose used for mapping the keypoint for tracking corrections

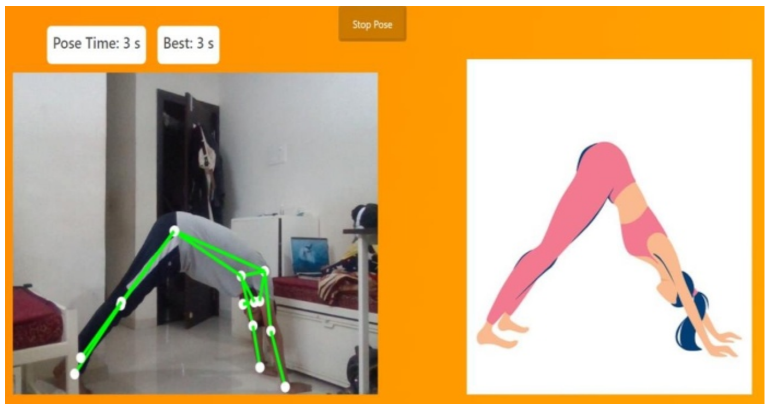


Fig. 21 Real time detection and tracking of **Dog Pose** (Left) real time yoga pose (Right) image ofcorrect Dog pose used for mapping the keypoint for tracking corrections

4. Conclusions

The previous few years have seen a lot of study into determining where a person is. Different estimates of human posture compared to other computer vision difficulties in that it has to find and construct human body components using an existing human bodily composition. Yoga posture estimates used for fitness may boost a person's safety and prevent injuries effectiveness.

The work proposed aYoga pose detection using Movenet architecture. To improve accuracy a CNN layer was added initially to the Movenet architecture. The proposed method demonstrates an accuracy of around 90%. For real time implementation of the Yoga pose detection and tracking, an application is created using React.

The postures used in yoga can be examined using image analysis to determine the correctness. We suggest that a self-learning yoga programme can both make yoga more widely accepted and guarantee that it is done properly. The tremendous research being done in this area makes deep learning techniques appealing. All eight Surya Namaskar yoga postures are accurately classified when hybrid CNN and LSTM models are applied to Open Pose data. Native CNNs and SVMs can also be used, however SVMs do not perform well on huge datasets. Compared to models built on CNN + LSTM, this is less effective. As a result, without a mentor, the proposed method will enable participants to correctly and successfully practice the Yoga posture.

References

[1] Si J, Jiang F and Shen R (2019) Large-pose Face Alignment via Shape-aware Heatmap. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 3037-3041.

[2] Sunney J, Jilani M, Pathak P, Stynes P (2023) A Real-time Machine Learning Framework for Smart Home-based Yoga Teaching System, International Conference on Machine Vision and Information Technology (CMVIT).

[3] Chaudhari A, Dalvi O, Ramade O and Ambawade D (2021) Yog-Guru: Real-Time Yoga Pose Correction System Using Deep Learning Methods International Conference on Communication information and Computing Technology (ICCICT), Mumbai, India,1-6.

[4] Pala VCR, Kamatagi S, Jangiti S, Swaraja K, Madhavi K and Kumar G (2023) Yoga Pose Recognition with Real time Correction using Deep Learning International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 387-393.

[5] X. Huang, D. Pan, Y. Huang, J. Deng, P. Zhu, P. Shi et al. (2021) Intelligent Yoga Coaching System Based on Posture Recognition International Conference on Culture-Oriented Science and Technology, Communication University of China, School of Information and Communication Engineering, Beijing, China, 290–293.

[6] Yingdong R (2022) Research on Different Convolutional Neural Networks in the Classification Scenes of Yoga Poses based on OpenPose Extraction. International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Dalian, China 1532-1535.

[7] Goel S, Mohanty S, Markanday S (2022) Classification of Yoga Pose using Pre-trained CNN Models and Machine Learning Classifier. International Conference on Current Development in Engineering and Technology (CCET), Bhopal, India, 1-6.

[8] Paula P and William S (2018) Machine Learning Gesture Analysis of Yoga for Exergame Development, IET Cyber-Physical Systems: Theory & Applications, 3(2):106-110.

[9] Jun J, Lee J, and Kim C (2020) Human Pose Estimation Using Skeletal Heatmaps Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Auckland, New Zealand, 1287-1292.

[10] Zhou Y, Jiang G and Lin Y (2016) A novel finger and hand pose estimation technique for real-time hand gesture recognition. Pattern Recognition, 49:102-114.

[11] Bajpai R and Joshi D (2021) MoveNet: A Deep Neural Network for Joint Profile Prediction Across Variable Walking Speeds and Slopes. IEEE Transactions on Instrumentation and Measurement 70:1-11.

[12] Zixin C, Fernando O and Ong J (2022) PoseBuddy: Pose Estimation Workout Mobile Application, 2022 International Conference on Cyberworlds (CW), Kanazawa, Japan, 151-154.

[13] Bejinariu S, Luca R, Costin H, Rotaru F, Onu I (2022) Automatic Gait Analysis and Classification in Video Sequences. International Conference and Exposition on Electrical And Power Engineering (EPE), Iasi, Romania, 031-036.

[14] Sathyanarayanan G, Vengadavaradan A and Bharadwaj B, (2019) Role of Yoga and Mindfulness in Severe Mental Illnesses: A Narrative Review. International journal of Yoga, 12(1): 3-28.

[15] Chen H, He Y, Hsu C, Chou C, Lee S and Lin B (2014) Yoga posture recognition for self- training", Lecture Notes in Computer Science, 8325,PART 1: 496-505.

[16] Islam M, Mahmud H, Ashraf F, Hossain I and Hasan M (2017) Yoga posture recognition by detecting human joint points in real time using microsoft kinect. IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Dhaka, 668-673.

[17] Yadav S, Singh A, Gupta A, Raheja J (2019) Real-time Yoga recognition using deep learning. Neural Computing & Applications. 31: 9349-9361.

[18] Chena Y, Tianb Y and Hea M (2020) Monocular Human Pose Estimation: A Survey of Deep Learning-based Methods. Computer Vision and Image Understanding, 192.

[19] Osokin D (2019) Real-time 2D Multi-Person Pose Estimation on CPU: Lightweight OpenPose, Proceedings of the International Conference on Pattern Recognition Applications and Methods, 744-748.

[20] Chen H, He Y and Hsu C (2018) Computer Assisted Yoga Training System. Multimedia Tools and Application., 77(18): 23969-23991.

[21] Andriluka M,Iqbal M, Insafutdinov E, Pishchulin L, Milan A, Gall J and Schiele B (2018) PoseTrack: A Benchmark for Human Pose Estimation and Tracking, International Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 5167-5176.

[22] Yang W, Ouyang W, Li H and Wang X (2016) End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation International Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 3073- 3082.

[23] Pishchulin L, Insafutdinov E, Tang S, Andres B, Andriluka M, Gehler P et al. (2016) Deepcut: Joint subset partition and labeling for multi person pose estimation. IEEE Conference on Computer Vision and Pattern Recognition.

[24] Downs H, Miltenberger R, Biedronski J and Witherspoon L (2015) The Effects of Video Self-Evaluation on Skill Acquisition with Yoga Postures. Journal of applied behavior analysis, 484: 930-935.

[25] Lee J, Kim D, Jeong S, Choi S (2018) Diagnosis and prediction of periodontally compromised teeth using a deep learning-based convolutional neural network algorithm. Journal of Periodontal Implant Sci. 48(2):114-123.